Treed Regression

William P. Alexander and Scott D. Grimshaw

Given a data set consisting of n observations on p independent variables and a single dependent variable, treed regression creates a binary tree with a simple linear regression function at each of the leaves. Each node of the tree consists of an inequality condition on one of the independent variables. The tree is generated from the training data by a recursive partitioning algorithm. Treed regression models are more parsimonious than CART models because there are fewer splits. Additionally, monotonicity in some or all of the variables can be imposed.

Key Words: CART; MARS; Nonlinear regression models; Recursive partitioning; Tree-structured regression.

1. INTRODUCTION

1.1 Motivation

In regression analysis, a dependent variable, Y, is modeled as a function of one or more independent variables, $X(1), \ldots, X(p)$. The relationship can be expressed as

$$Y_j = f\Big(X_j(1), X_j(2), \ldots, X_j(p)\Big) + \epsilon_j, \quad j = 1, \ldots, n,$$

where $f(\cdot)$ is the deterministic component and the ϵ_j's are the random component, which are uncorrelated with mean 0 and variance σ^2.

Consider the regression model specified in Figure 1. It consists of a binary tree with generic regression functions at the leaves. Each node of the tree consists of an inequality condition on one of the independent variables. This model encompasses many well-known regression methodologies as special cases. For example, linear regression does not split the population; that is, the root node is the one and only leaf. The entire sample is modeled using a single regression function

$$f_1(\mathbf{X}; \theta_1) = \theta_{10} + \theta_{11}X(1) + \cdots + \theta_{1p}X(p).$$

CART, proposed by Breiman, Friedman, Olshen, and Stone (1984), is also a special case of the Figure 1 model. The CART model is realized by retaining the full tree structure

William P. Alexander is Director of Quantitative Program Research, NBC Program Research, 3000 West Alameda Avenue, Burbank, CA 91523; e-mail: william.alexander@nbc.com. Scott D. Grimshaw is Assistant Professor, Department of Statistics, Brigham Young University, Provo, UT 84602; e-mail: grimshaw@byu.edu.

Journal of Computational and Graphical Statistics, Volume 5, Number 2, Pages 156–175

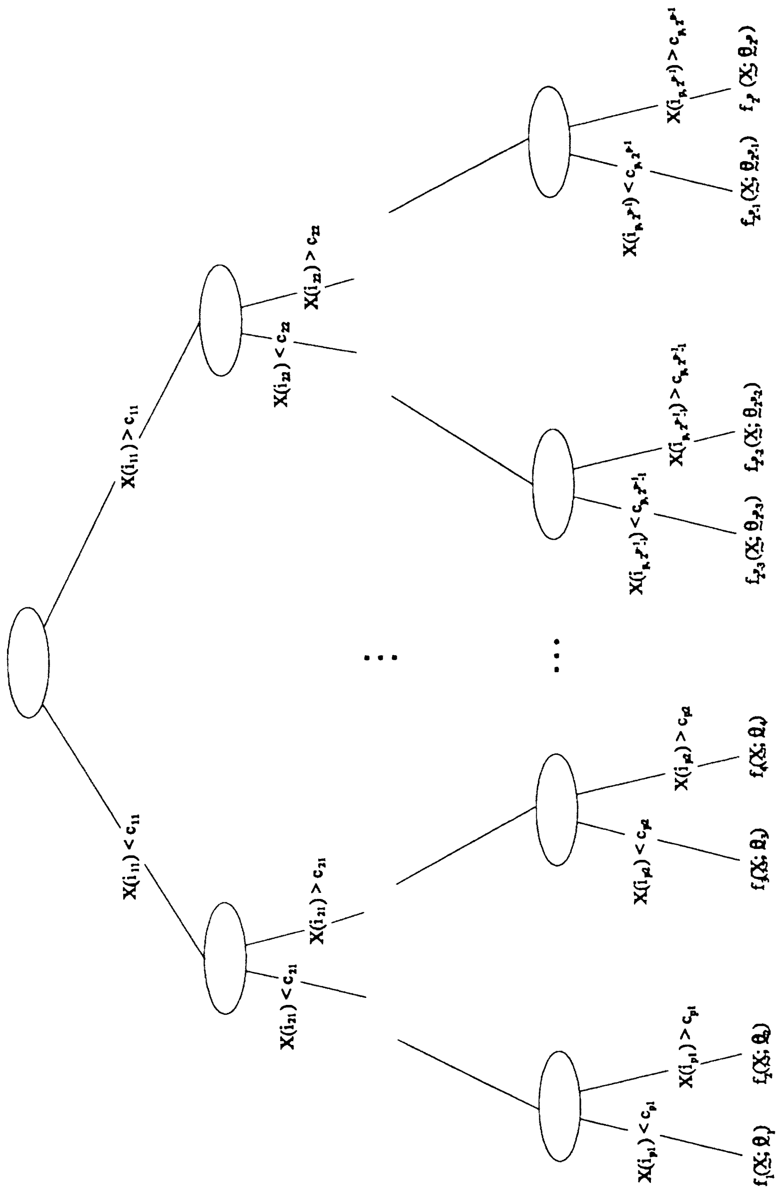


Figure 1. General Tree-Structured Regression Model.

and specifying the regression functions to be constants; that is,

$$f_i(\mathbf{X}; \theta_i) = \theta_i.$$

Treed regression proposes a compromise between linear regression, with no tree structure but one complex linear model, and CART, with a full tree structure and a very simple model at each leaf. The treed regression model retains the full tree structure of the Figure 1 model, but specifies the leaf models as *simple linear regressions,* where

$$f_i(\mathbf{X}; \theta_i) = \theta_{i1} + \theta_{i2} X(k_i).$$

Providing a more complex model at the leaves takes advantage of the linear structure found often in practice. This can be expected to lead to trees of lesser depth, which are generally more interpretable. The partitions induced by the tree structure may be interpretable by the researcher in their own right as groups understood to have a commonality of behavior. The structure can also be interpreted as thresholds. For instance, in a physical system if the temperature and pressure exceed certain thresholds then the system produces output according to one relation. Should these conditions not be met, then another relationship holds.

1.2 Relation to Other Technologies

Treed regression is distinct from switching regressions and change point analysis. The goal of these latter two is to identify a specific point(s) at which the parameters of a full model change. See Zacks (1991) for a review.

There are scattered references in the literature to segmenting regressions in an algorithmic, tree-structured fashion. The goals, technologies, and types of models placed at the leaves differ from treed regression. Duarte and Kalff (1990) presented an example in which a CART model of a single split is created, then the observations in each of the two subsets are modeled independently with linear models. Breiman and Meisel (1976) and Karalič (1992) both suggested fitting full regression models at the leaves. Breiman and Meisel's goal was to arrive at an estimate of the error variance requiring the least assumptions in order to assess the goodness-of-fit of the original regression. The splitting is driven by a criterion based on an F-statistic. Karalič (1992) described the changes that must be made to CART to permit linear models at the leaves. He employed a Bayesian splitting criterion.

Friedman (1991) proposed MARS, which combines a generalization of the recursive partitioning methodology and spline fitting. The MARS model is highly adaptive, continuous, and capable of excellent local approximations to complicated functions in high dimensions. In its most commonly used form, the MARS model is a sum of a product of regression coefficients and basis functions written as

$$f(\mathbf{X}; \beta, t) = \beta_0 + \sum_k \beta_k \prod_{j=1}^{n_k} [\pm(X(\nu_{jk}) - t_{jk})]^+,$$

where the basis functions are first-order truncated power splines defined by optimally selected independent variables, $X(\nu_{jk})$, and knot locations, t_{jk}. The MARS algorithm selects the number of basis functions as well as the independent variables and knot locations. Friedman points out that the binary tree-structure is a special case of the basis functions used in MARS in which each binary split creates two basis functions. However, this generalization abandons the tree structure's appealing decision criterion and it becomes far more difficult to interpret an independent variable's effect from the functional specification.

Chaudhuri, Huang, Loh, and Yao (1994) correctly identified the advantages of employing a decision tree created by recursive partitioning to provide information about the regression variables with simple and functionally explicit models at the leaves. They suggest fitting polynomial or full regression models at the leaves but recognize the extensive computational burden in evaluating all possible multiple regression models for all possible decision trees. To significantly reduce this intensive search, their technique evaluates the observations associated with a fitted model's positive and negative residuals for each $X(i)$. If the fit is adequate both sets should be similar and no division of the data is required. For instance, suppose $Y = X^2$ for $-1 < X < 1$ and a straight line is fit to the data. The variance of the X's associated with positive residuals would be significantly larger than the variance of the X's associated with negative residuals. Chaudhuri, Huang, Loh, and Yao (1994) used this as evidence to indicate that the fit could be improved by splitting the data.

One could also consider performing a variable selection for the linear models or more general leaf regression models like polynomials. However, the computational burden involved is overwhelming for current resources. To be feasible, the number of candidate models must be limited in some fashion. Also, issues such as collinearity, model interpretability, and over-parameterization become increasingly critical as leaf models grow in complexity. Treed regression maintains a feasible problem by limiting itself to the class of simple linear regressions.

2. ALGORITHM

2.1 Overview

The treed regression algorithm consists of a base algorithm that is applied recursively. The base algorithm generates a tree with a single node and two leaves. The node divides the original data set into two mutually exclusive data sets. The base algorithm can then be applied to each of these two data sets, generating a tree with four leaves. In this manner, a tree of any depth can be generated.

An implementation of this algorithm is available in C and S from the authors.

2.2 Algorithm Details

The base algorithm is defined below:

1. For $i = 1, \ldots, p$ do:
   Evaluate the independent variable $X(i)$ as the variable to use in the node inequality. This is done as follows:

2. Sort the $X(i)$'s into the unique, ascending values $Z_1, \ldots, Z_m$, $(m \leq n)$. The candidate cutpoints for the node are given by

$$c_j = \frac{Z_j + Z_{j+1}}{2},$$

where $j = 1, \ldots, m-1$.
3. For $j = 1, \ldots, m-1$ do the following:
The candidate split, c_j, is rejected immediately if one of the leaves contains fewer than t observations. The software defaults to a value of $t = 10$.
4. The original data set is divided into two new data sets. The kth observation is assigned to the left leaf if the kth value of $X(i)$ is less than c_j and to the right otherwise.
5. For each leaf, each independent variable is evaluated as the regressor variable. The best linear regression is determined for each leaf independently. "Best" is defined as least sum of squared error.
6. The sum of squared error for the node consisting of splitting $X(i)$ at c_j is the sum of the SSEs of the left and right leaves.
7. If this value of the node SSE is the best observed, the values defining the model (node variable, cutpoint, regression variables, and regression parameter values) are retained.

The base algorithm finds the optimal tree of depth 1 (one node and two leaves). To generate a tree of depth two, the node, which splits the original data set into two new data sets, is retained. The base algorithm is then applied to each of these data sets. The original regressions from the first application of the base algorithm are discarded. If there are fewer than $2t$ observations in one of these data sets, then no further splitting occurs down that branch and the existing regression is retained. This implies that sample sizes of 50 and larger are usually required to build trees of moderate size. There is an implicit variable selection performed in choosing the "best" split variable and the "best" regressors in the terminal nodes.

2.3 COMPUTATIONAL EFFICIENCY

The base algorithm requires that approximately $2np^2$ regressions be computed. To improve the computational efficiency, these regressions can be organized in such a way that observation update formulas can be used. As the candidate values of c_i are tried in succession, observations will move from the right leaf to the left leaf. At each iteration, all possible p linear regressions must be evaluated for each leaf. An efficient way is to maintain and update the parameter values and sum of squared errors for each of the p candidate regressions for each leaf.

Because simple linear regression models are used at each leaf, the first approach is to maintain the sums and sums of squares and cross-products required for simple linear regressions based on each independent variable for each leaf. These quantities are updated as observations move from right to left as the cutpoint changes. This technique requires minimal storage, but one should center and scale the raw data to avoid potential round-off error problems. This approach is used in the algorithm available from the authors.

The second approach maintains, for each leaf, a QR decomposition for each regressor. The Householder transformation is used for the initial decomposition for each leaf and the Givens transformation is used to update the decomposition as observations are added or deleted from the leaves. This approach, which decomposes the matrix X, is more computationally stable than methods that form the $X'X$ matrix. Also, because this approach applies to a linear model with p independent variables, it would be preferred if the algorithm were generalized to permit polynomial or multiple regression models at the leaves. See Seber (1977) for details on the QR decomposition and updating and Alexander and Grimshaw (1994) for details on applying it to the treed regression algorithm.

3. COMPARISON TO OTHER REGRESSION METHODOLOGIES

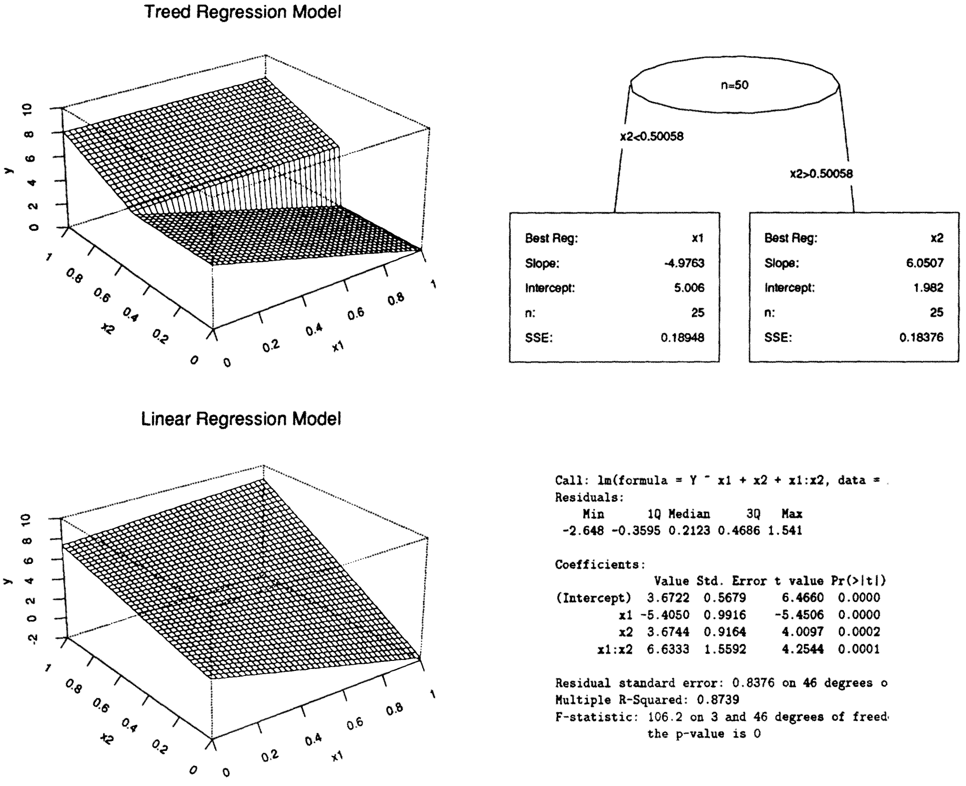
Linear regression models are extremely popular because they are elegant in theory and highly effective in practice since many relationships display nearly linear behavior over a given domain. If the specification of the linear regression model masks the presence of a nonlinear effect, however, the user may miss valuable information. A tree structure, as employed by CART, is an attractive way to convey the effect of the independent variables. CART models are especially appealing for less-mathematically inclined users. Unfortunately, the step function approximation to many functions is quite poor, and CART may construct a large tree structure in an effort to improve the quality of the approximation to a smooth function. By concentrating on point-wise prediction, many modern regression models are far more flexible in approximating more complicated surfaces. These methods either have no explicit model expression (kernel smoothers) or rely on an approximation to the underlying function whose functional form is not usually of interest (smoothing splines and MARS). This can be frustrating for users interested in interpreting the effect of the independent variables on the dependent variable.

In this section, three examples are used to compare treed regression to linear regression, CART, and MARS. The underlying function of Example 1 has a threshold effect in which there is a different relationship above and below the threshold. If the threshold effect were known, a linear model could be modified to perform quite well, but in these examples the model is intended to diagnose this effect. Suppose the true relationship is

$$Y = \begin{cases} 5 - 5X(1) + \epsilon, & \text{if } X(2) < \frac{1}{2} \\ 2 + 6X(2) + \epsilon, & \text{if } X(2) > \frac{1}{2} \end{cases},$$

where $\epsilon \sim N(0, .1)$ and $[X(1), X(2)]$ are distributed uniformly on the unit square. Figure 2 contains plots of the surface for the treed regression, linear regression, CART, and MARS models constructed from a sample of $n = 50$. The model is provided in each case except MARS.

Treed regression will clearly perform well for this relationship because the threshold effect is a binary partition and the two subspaces each contain a simple linear regression model. In comparison, the linear regression model provides an apparently acceptable fit according to hypothesis tests and R^2. Unfortunately, the model smooths over the threshold masking the effect of $X(1)$ on $X(2) < \frac{1}{2}$. This model is quite poor for $X(1) >$



Best Reg:	x1
Slope:	-4.9763
Intercept:	5.006
n:	25
SSE:	0.18948

Best Reg:	x2
Slope:	6.0507
Intercept:	1.982
n:	25
SSE:	0.18376

```
Call: lm(formula = Y ~ x1 + x2 + x1:x2, data = .
Residuals:
    Min      1Q Median     3Q   Max
 -2.648 -0.3595 0.2123 0.4686 1.541

Coefficients:
               Value Std. Error t value Pr(>|t|)
(Intercept)  3.6722  0.5679      6.4660  0.0000
         x1 -5.4050  0.9916     -5.4506  0.0000
         x2  3.6744  0.9164      4.0097  0.0002
      x1:x2  6.6333  1.5592      4.2544  0.0001

Residual standard error: 0.8376 on 46 degrees o
Multiple R-Squared: 0.8739
F-statistic: 106.2 on 3 and 46 degrees of freed
             the p-value is 0
```

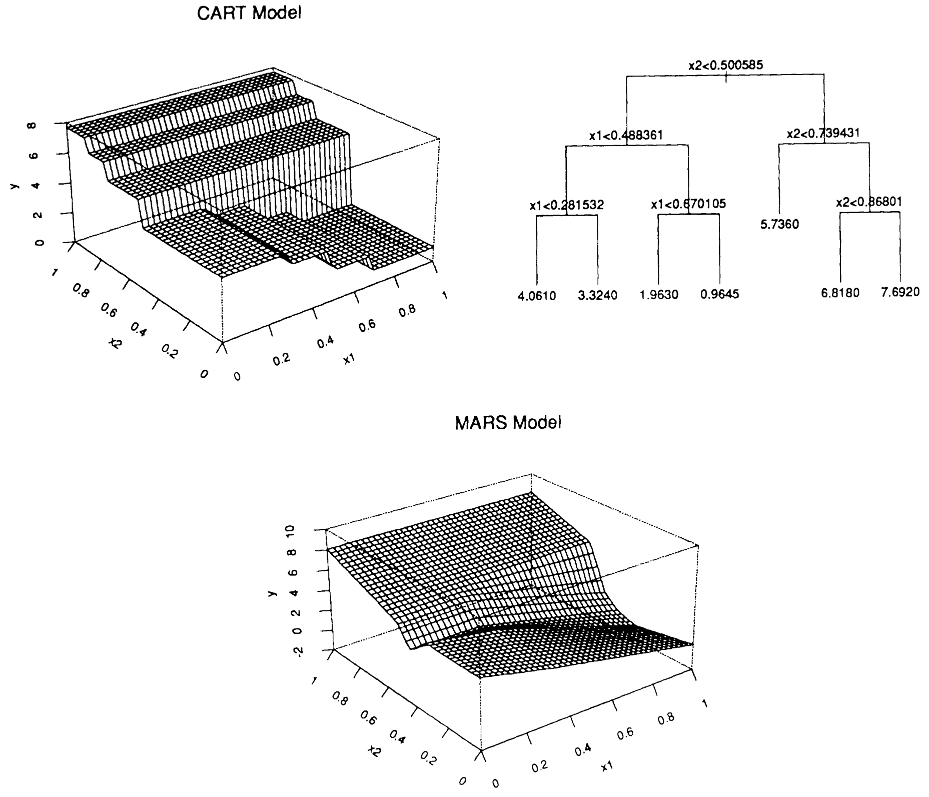
Figure 2. Example 1 contains a threshold effect, which is modeled using treed regression (top) and linear regression (bottom).

$\frac{1}{2} \cap X(2) < \frac{1}{2}$. The CART model identifies the threshold effect with the first split in the tree structure. However, the remaining five split nodes do not represent other threshold effects, but are necessary to approximate the linear components with a step function. The MARS model performs well at the corners of the domain, but since the MARS model is continuous the threshold effect is smoothed over.

The mean integrated square error (MISE) was approximated by simulation to compare the regression methods. Each regression method was used on each of 100 samples, and for each iteration the MISE was approximated by Monte Carlo integration using 100 replications generated from the true relationship. Table 1 contains the MISE for Example 1. As anticipated, treed regression provides the best fit. The MARS model, which imposes continuity when it does not exist, gives the next best fit followed by the CART model, which creates a discontinuous step function. The linear regression model performs worst, which is expected since it smooths over the threshold effect.

Example 2 also contains a threshold effect, but in this case the surface is continuous at the threshold. Suppose the true relationship is

$$Y = \begin{cases} -2 + 4X(1) + 5X(2) + \epsilon, & \text{if } X(1) < \frac{1}{2} \\ 2 - 4X(1) + 5X(2) + \epsilon, & \text{if } X(1) > \frac{1}{2} \end{cases},$$


CART Model

MARS Model

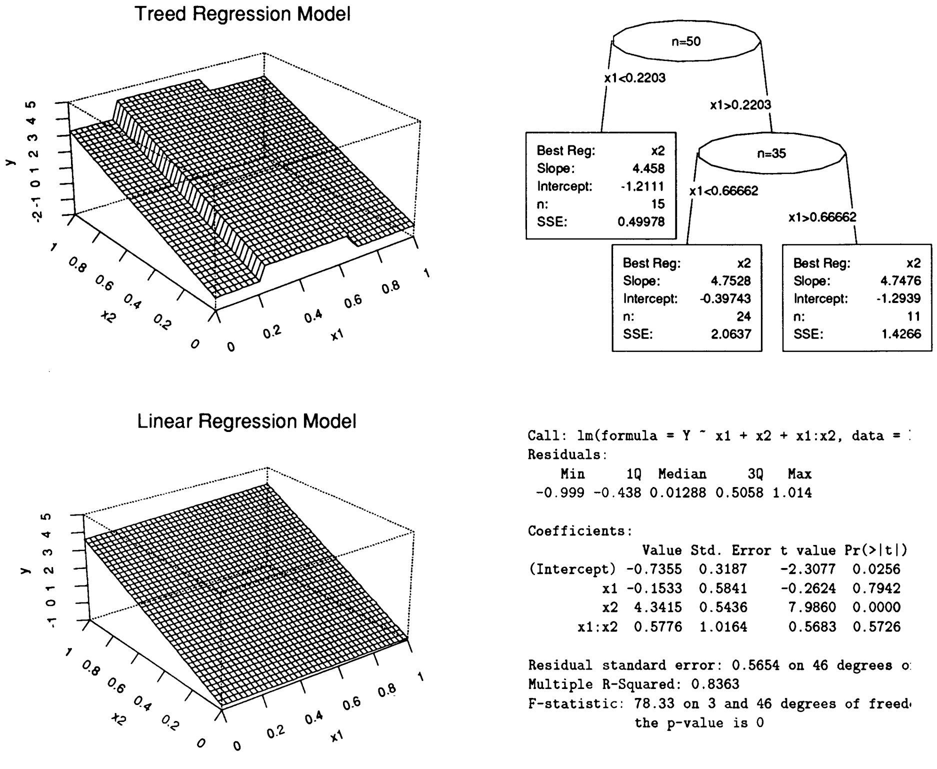
Figure 2. (continued) Example 1 contains a threshold effect, which is modeled using CART (top) and MARS (bottom).

where $\epsilon \sim \mathrm{N}(0, .1)$ and $[X(1), X(2)]$ are distributed uniformly on the unit square. Notice that treed regression is not expected to perform quite as well as in Example 1 because there are multiple regression models above and below the threshold. Figure 3 contains plots of the surface for the treed regression, linear regression, CART, and MARS models constructed from a sample of $n = 50$. The model is provided in each case except MARS.

The MARS model provides an excellent approximation to this surface. Because MARS provides a continuous model, it captures this threshold effect nicely. The treed regression model creates three terminal nodes, each regressing on $X(2)$ with nearly equal slopes but different intercepts to approximate the surface. The split nodes in the tree structure represent the threshold effect of $X(1)$. Both the linear regression and CART models effectively ignore the threshold effect. In the linear regression model, $X(1)$ is not significant and the model concentrates on the more dominant effect of $X(2)$. In the

Table 1. MISE for Example 1

Treed regression	*Linear regression*	*CART*	*MARS*
4.1725	38.4277	15.0585	8.9371



```
Call: lm(formula = Y ~ x1 + x2 + x1:x2, data = :
Residuals:
    Min     1Q  Median     3Q   Max
 -0.999 -0.438 0.01288 0.5058 1.014

Coefficients:
               Value Std. Error t value Pr(>|t|)
(Intercept) -0.7355  0.3187    -2.3077  0.0256
         x1 -0.1533  0.5841    -0.2624  0.7942
         x2  4.3415  0.5436     7.9860  0.0000
      x1:x2  0.5776  1.0164     0.5683  0.5726

Residual standard error: 0.5654 on 46 degrees o
Multiple R-Squared: 0.8363
F-statistic: 78.33 on 3 and 46 degrees of freed
           the p-value is 0
```

Figure 3. Example 2 contains a continuous threshold effect, which is modeled using treed regression (top) and linear regression (bottom).

CART model, the largest increases in the deviance function used to select optimal binary splits are in $X(2)$ since it has the dominant slope. Again, the CART model is a poor approximation because it results in a step function.

Table 2 contains the MISE for Example 2 approximated by simulation. As seen in the sample, MARS provides an excellent fit to the underlying relationship. However, treed regression performs competitively, especially compared to the linear regression and CART models. The linear regression model performs poorly because it fails to capture the threshold effect. When the underlying relationship is smooth, as in this example, the CART model will perform poorly because of its discontinuous nature.

Example 3 was selected to have no threshold effects in order to investigate how methods that are based on recursive partitioning approximate a linear function. Suppose the true relationship is

$$Y = 5 + 10X(1) - 5X(2) - 5X(1) \cdot X(2) + \epsilon,$$

where $\epsilon \sim \mathrm{N}(0, .1)$ and $[X(1), X(2)]$ are distributed uniformly on the unit square. Figure 4 contains plots of the surface for the treed regression, linear regression, CART, and MARS models constructed from a sample of $n = 100$. The model is provided in each case except MARS.

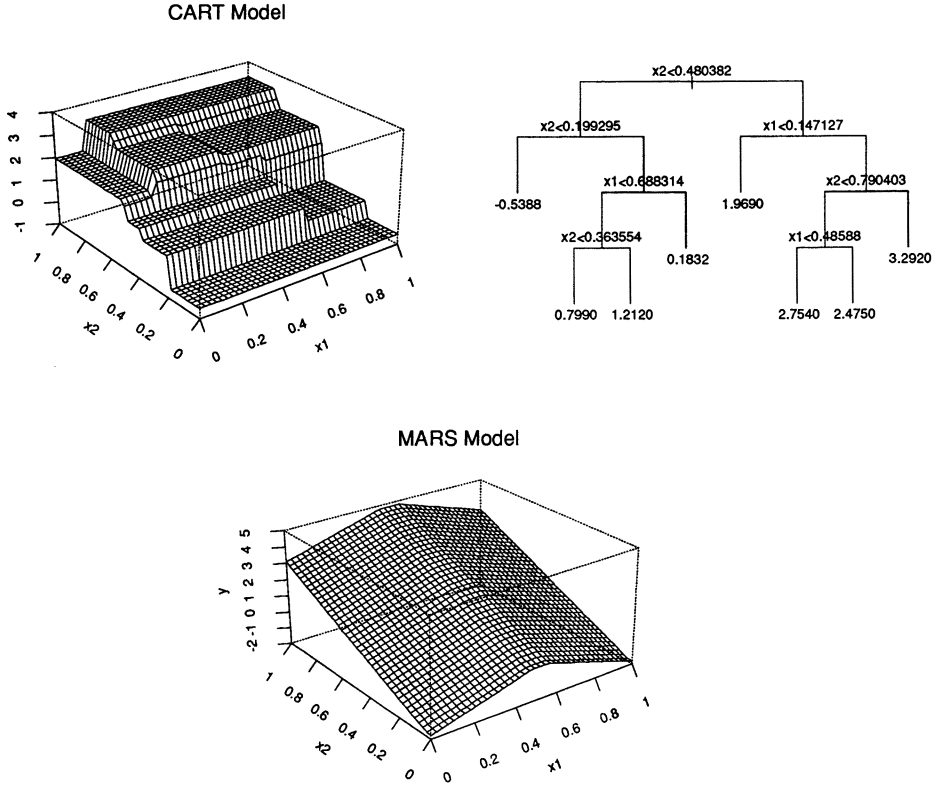


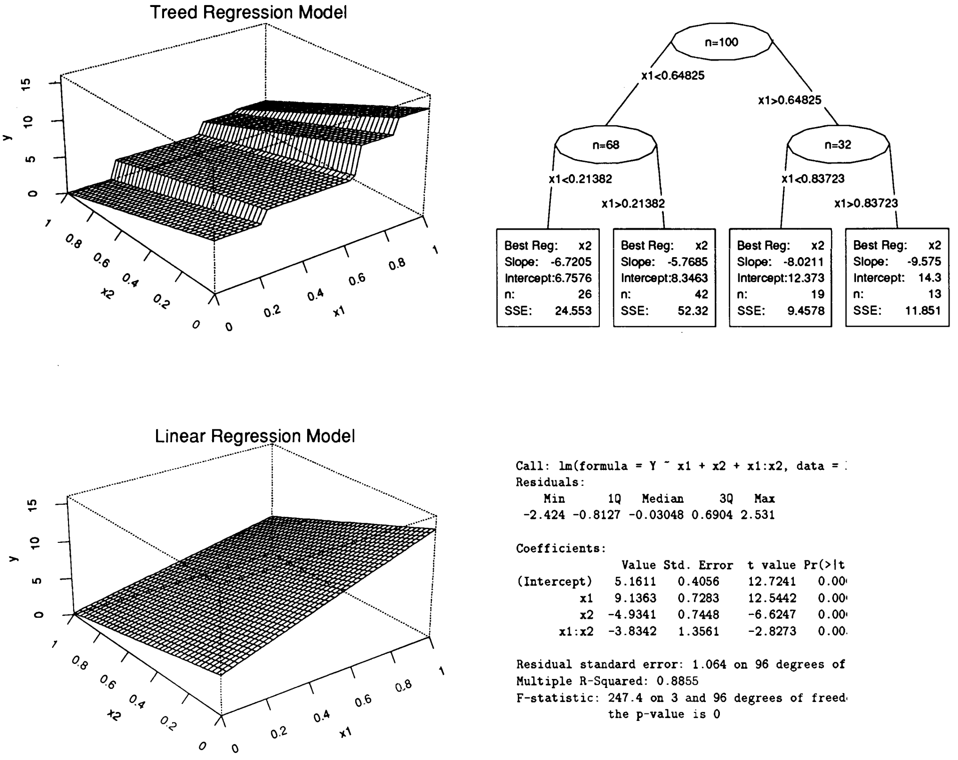
Figure 3. (continued) Example 2 contains a continuous threshold effect, which is modeled using CART (top) and MARS (bottom).

Obviously the linear regression model will yield the superior fit. With a correctly specified functional form and only four estimated parameters it is the most efficient use of the data. The MARS model is nearly equivalent except for a small bend where MARS interpreted the noise to represent a threshold effect. CART is inefficient in this case, where 15 split nodes are created yielding 16 estimated terminal node means in an effort to approximate the underlying relationship. The treed regression model, in contrast, creates a small tree structure with four terminal nodes. The simple linear regression model at each of the four terminal nodes are better able to approximate the underlying relationship than the CART model.

Table 3 contains the MISE for Example 3 approximated by simulation. The linear regression model provides the best fit, as was expected. The MARS model was penalized slightly by its adaptive nature in this example. Occasionally, when there was insufficient

Table 2. MISE for Example 2

Treed regression	*Linear regression*	*CART*	*MARS*
5.3899	18.3472	22.3600	.2374



```
Call: lm(formula = Y ~ x1 + x2 + x1:x2, data = :
Residuals:
    Min      1Q   Median      3Q   Max
 -2.424 -0.8127 -0.03048 0.6904 2.531

Coefficients:
               Value Std. Error  t value Pr(>|t
(Intercept)   5.1611    0.4056   12.7241   0.00
         x1   9.1363    0.7283   12.5442   0.00
         x2  -4.9341    0.7448   -6.6247   0.00
      x1:x2  -3.8342    1.3561   -2.8273   0.00

Residual standard error: 1.064 on 96 degrees of
Multiple R-Squared: 0.8855
F-statistic: 247.4 on 3 and 96 degrees of freed
             the p-value is 0
```

Figure 4. Example 3 is a plane with no threshold effects, which is modeled using treed regression (top) and linear regression (bottom).

data in a given domain to clearly separate the functional relationship from the noise, MARS would model the noise. CART is clearly the most inefficient. As was seen in the previous example, when the underlying relationship is smooth, CART creates a large tree with many terminal nodes to obtain a step function approximation. Treed regression can be viewed as an improvement to CART by permitting a slightly more complicated model at the terminal nodes which results in a smaller tree structure and better approximation to the underlying relationship. Although the quality of the approximation may not equal the performance of MARS, it is suggested that treed regression has other desirable features which are worth the small loss in MISE.

In summary, the three examples demonstrate the performance of treed regression compared to other regression methodologies. When a threshold effect is present, treed regression performs comparably to MARS and is superior to linear regression, where

Table 3. MISE for Example 3

Treed regression	*Linear regression*	*CART*	*MARS*
52.8244	4.0373	118.8893	20.4207

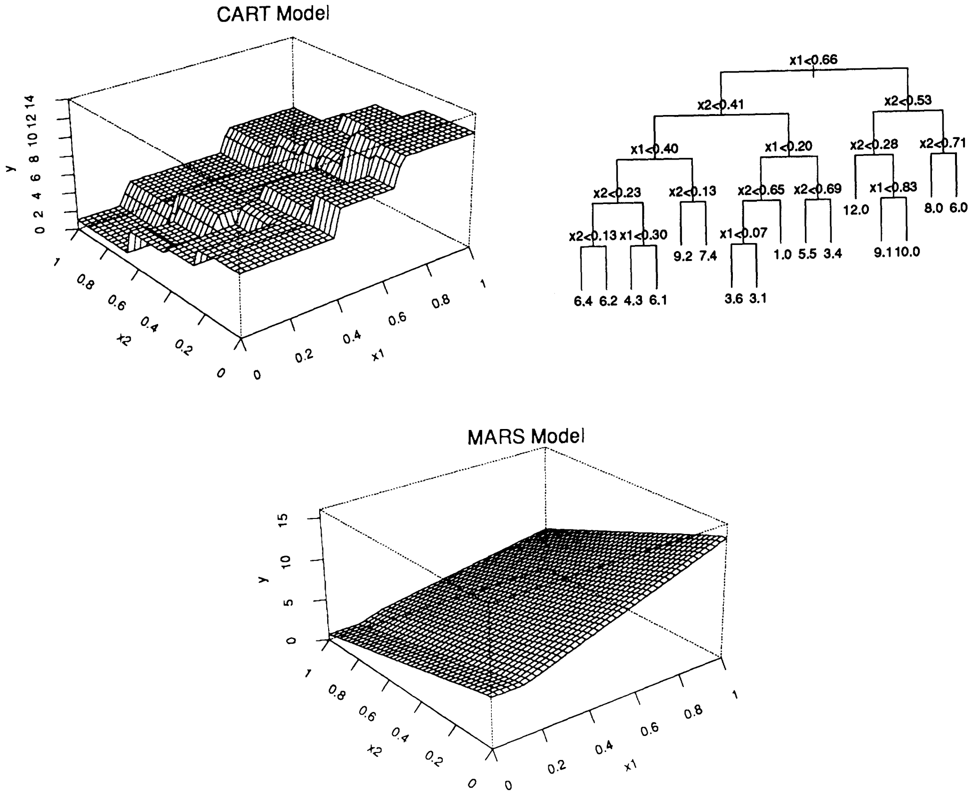


Figure 4. (continued) Example 3 is a plane with no threshold effects, which is modeled using CART (top) and MARS (bottom).

the threshold effect is ignored, and to CART, where the threshold effect is included in the tree structure with other splits created to approximate the underlying relationship with a step function. When the underlying relationship is smooth, treed regression is a significant improvement over CART, at least in terms of prediction, because it yields a smaller tree structure with locally linear fits at the terminal nodes. Although MARS is more effective for accurate prediction of a smooth surface, treed regression may provide the data analyst with the advantage in understanding and interpreting the effect of the independent variables because of its simple form.

4. MONOTONICITY

In the linear regression model, monotonicity is not usually imposed as part of parameter estimation. Instead, the parameter estimates are evaluated to determine whether expected relationships between the mean of Y and $X(i)$ hold. However, it is far more difficult to discern whether monotonicity in $X(i)$ is achieved for tree-based models . Therefore, if monotonicity is expected in some or all of the independent variables in the treed regression model, it is better to impose it by modifying the algorithm described in

Section 2. One then has a model guaranteed to be monotone, whose fit can be compared to the unrestricted model.

Associate with the ith independent variable an indicator, π_i, which gives the direction of the response of the mean of Y to an increase in $X(i)$. Specifically, let

$$\pi_i = \begin{cases} -1 & \text{if the response is negative} \\ 0 & \text{if the response is not restricted} \\ 1 & \text{if the response is positive.} \end{cases}$$

A first step in modifying the algorithm imposes monotonicity *within* a given leaf. Step 5 of the base algorithm is modified to be:

5$'$. The best linear regression is determined for each leaf independently. "Best" is defined to mean least sum of squared error among those regressions that satisfy the directionality constraint. If no such regressions exist, a model with slope equal to 0 is used.

Monotonicity *between* leaves must also be addressed. Suppose the variable $X(i)$ is employed at the root node with a cut value of c^* and $\pi_i = 1$. As the value of $X(i)$ passes from $c^* - \epsilon$ to $c^* + \epsilon$, the tree output is generated by a different leaf regression. Without constraints, there is no guarantee that the tree output will not decline.

The monotone treed regression algorithm accounts for leaf-switching to maintain a truly monotone model. In the standard treed regression, the observation $(X(1), \ldots, X(p))$ will fall down the tree to a single leaf. In monotone treed regression, one not only examines this leaf but all leaves that can be reached by reducing any $X(i)$ value with $\pi_i = 1$ and increasing any $X(i)$ with $\pi_i = -1$. That is, one examines the terminal node regression models that would be used by making the observation "worse" in terms of the expected relation with $X(i)$. The set of output values of these leaves is constructed. The output of the tree is taken to be the maximum of this set.

The tree-building algorithm must be adjusted to take account of this new requirement for monotonicity. The algorithm builds the tree up from the leaf with the least precedence to highest precedence. Step 5 of the base algorithm becomes

5$''$. When calculating the SSE for a candidate regression, the fitted value is taken to be the maximum of the candidate regression fitted value and an input vector of fitted values containing the predicted value from all terminal regression models obtained by reducing $X(i)$ if $\pi = 1$ and increasing $X(i)$ if $\pi = -1$.

(a) If $\pi_i = 1$, then first determine the optimal regression for the left leaf. When determining the optimal regression for the right leaf, the fitted value for calculating SSE is taken as the maximum of the input fitted value, the fitted value from the left leaf model, and the candidate regression.

(b) If $\pi_i = -1$, then the procedure is the same except that one starts with right leaf and then moves to the left leaf.

(c) If $\pi_i = 0$, then the procedure is that described in Section 2.

A similar process continues in applying the base algorithm recursively. The leaf with least order is split first. When moving to the second leaf (the one with greater order) fitted values for these observations are first found from the monotone treed regression that was generated from the lesser leaf. These values are the input fitted values fed to the algorithm mentioned in 5$''$.

Table 4. Explanatory Variables Measured on Census Tracts in the Boston SMSA in 1970

Notation	*Definition*	π_j
crim	per capita crime rate by town	-1
zn	proportion of a town's residential land zoned for lots	0
indus	proportion of nonretail business acres per town	-1
chas	Charles River dummy variable with value 1 if tract bounds on the Charles River	0
noxsq	nitrogen oxide concentration (parts per hundred million) squared	-1
rm	average number of rooms squared	+1
age	proportion of owner-occupied units built prior to 1940	0
dis	logarithm of the weighted distances to five employment centers in the Boston region	0
rad	logarithm of index of accessibility to radial highways	0
tax	full-value property tax rate (per $10,000)	-1
ptratio	pupil-teacher ratio by town	-1
b	$(\text{Bk} - 0.63)^2$ where Bk is the proportion of blacks in the population	0
lstat	logarithm of the proportion of the population that is lower status	-1

The models for the lesser leaves are fixed before their possible effect on the models for greater leaves is determined. One can modify the algorithm to provide some feedback of the effect of the models for lesser leaves on greater leaves when determining the former. This greatly adds to the computational burden.

5. EXAMPLE

The relationship between treed regression, linear regression, CART, and MARS will be further explored via an examination of a specific data set. The data is from Belsley, Kuh, and Welsch (1980), who investigated a hedonic housing-price equation based on 506 census tracts in the Boston Standard Metropolitan Statistical Area in 1970. The data is available through StatLib at http://lib.stat.cmu.edu/. The dependent variable, Y, is the log of the median value of owner-occupied homes in each census tract in 1970. Table 4 lists the 13 explanatory variables measured for each census tract.

In their discussion, Belsley, Kuh, and Welsch (1980) uncovered problems with the least-squares model and pursued robust estimation and regression diagnostics. The diagnostics led them to conclude that

> While Boston comprises 131 census tracts of a total of 506 in the sample, it accounts for 40 of the 67 observations [on selected census tracts which exceeded diagnostic cutoffs]. While we did not explore the point further, one might speculate from this that central-city behavior differs systematically from that of the surrounding towns. A second general characteristic is that adjacent areas often have similar diagnostic magnitudes. ... Thus there appear to be potentially significant neighborhood effects on housing prices that have not been fully captured by this model.

A tree structure is well suited to modeling these neighborhood effects. The CART model in Figure 5, pruned by cross-validation to determine the number of nodes, is complex. Of

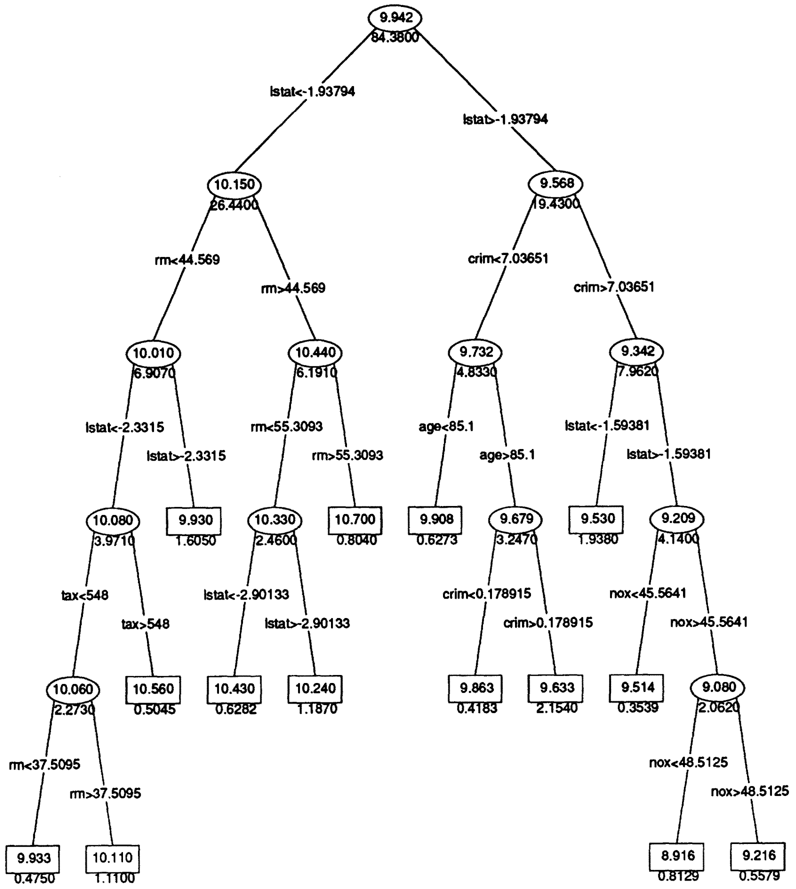


Figure 5. CART model of the log of the median value of owner-occupied homes using 13 explanatory variables measured on Boston area census tracts in 1970 from an example in Belsley, Kuh, and Welsch (1980).

the 14 nodes created by splits using six of the available explanatory variables, one must wonder how many characterize neighborhood effects and how many exist to compensate for inadequate functional modeling.

Consider "growing" a treed regression model that permits a small tree structure to characterize the neighborhood effects ignored by a multiple regression model and supplies simple linear regressions at the nodes to improve on the mean models of CART.

The model in Figure 6 is obtained by applying the base algorithm to the 506 census tracts. The proposed split on `tax` at 434.5 creates two interesting subsets. One subset contains all the census tracts from the city of Boston with the census tracts from the towns of Somerville and Chelsea, which are geographically close, and the single census tract containing the town of Middleton, which is quite removed. The second subset is comprised of the census tracts from towns surrounding downtown Boston which were developing suburbs.

Figure 7 demonstrates that most home prices in this suburban subset are well esti-

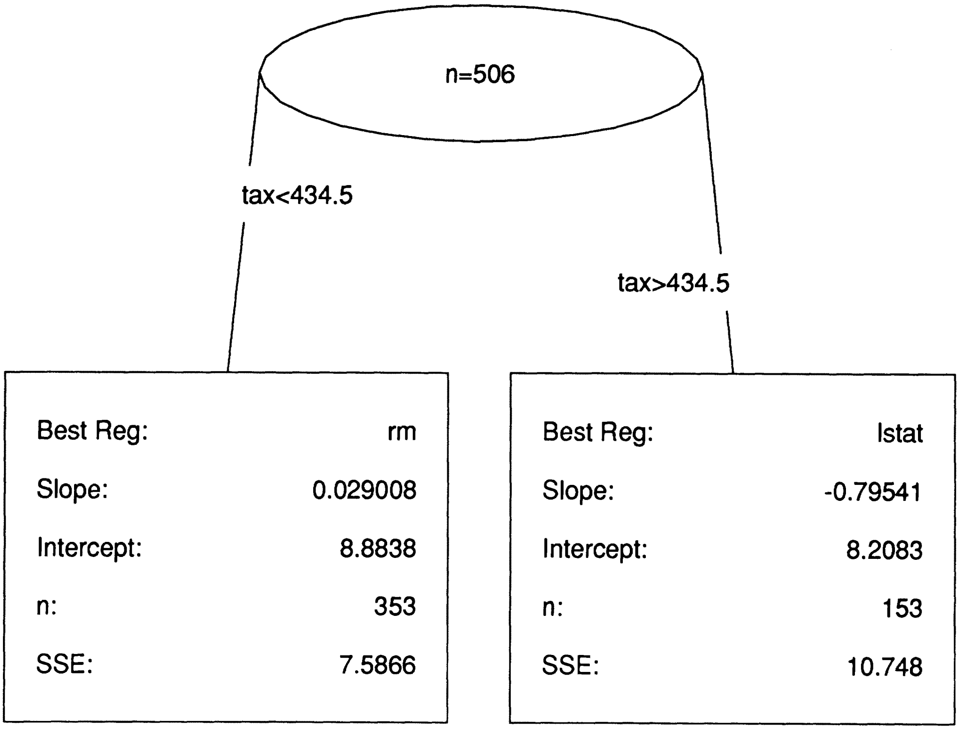


Figure 6. Treed regression model of the log of the median value of owner-occupied homes using 13 explanatory variables measured on Boston area census tracts in 1970 from an example in Belsley, Kuh, and Welsch (1980).

mated by a simple linear regression model using the variable rm, the average number of rooms squared. Furthermore, notice that the slope of the regression model for this node is consistent with the expectation that home price increases as the size of the home increases. The other node is also consistent with expectations. Whether or not the treed regression model is monotone between the segments defined using the independent variable tax requires more careful investigation. Consider the census tract labeled 371 in Belsley, Kuh, and Welsch (1980), which is in the Beacon Hill area of Boston. Because the value of tax is 666, the treed regression model predicts

$$\hat{Y} = 8.2083 - .79541\ \texttt{lstat} = 8.2083 - (.79541) \cdot (-3.519981) = 11.00813.$$

Consider, however, the prediction for the same census tract if the property tax rate were *decreased* enough to use the other branch of the treed regression model, all else equal, yielding

$$\hat{Y} = 8.8838 + .029008\ \texttt{rm} = 8.8838 + (.029008) \cdot (49.22426) = 10.31170.$$

Notice that a *decrease* in the tax rate would result in a *decrease* in the predicted home value, which is contrary to what is expected. This census tract is not unique since 15 of the 506 census tracts display this same feature.

The expected relationships between the dependent and independent variables can be maintained by applying the base algorithm, modified as described in Section 4, to impose

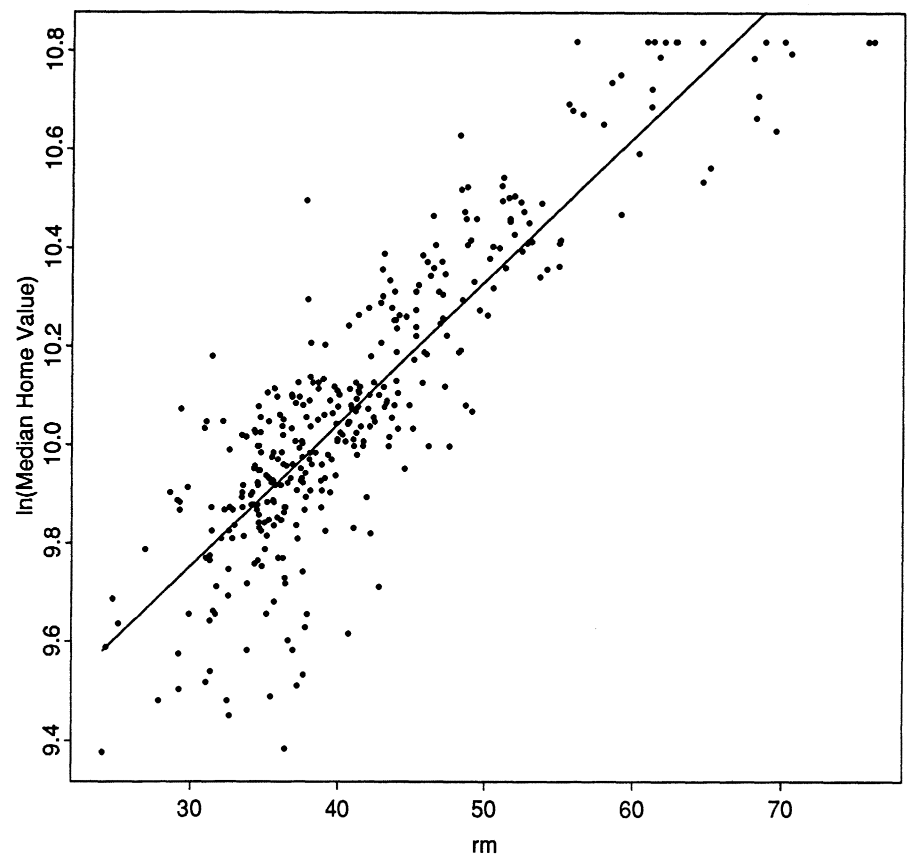


Figure 7. Simple linear regression model for the log of the median value of owner-occupied homes using rm, the average number of rooms squared, for census tracts satisfying tax < 434.5, which is the subset containing the cities of Boston, Somerville, Chelsea, and Middleton.

monotonicity in some the explanatory variables. The values π_i for each explanatory variable are given in Table 4 and the *monotone* treed regression model imposing these constraints is given in Figure 8. Notice that the monotone model creates two subsets based on rad. Even though a different explanatory variable is used in making the split, the subsets are very similar to those from the non-monotone treed model previously discussed because tax and rad are highly correlated. The division is cleaner and divides the sample into the census tracts in the city of Boston and the census tracts in the Boston suburbs.

The monotonicity constraint does have a slight cost. The treed regression model in Figure 6 has an SSE $= \sum(Y-\hat{Y})^2 = 18.3346$, but the monotone treed regression model in Figure 7 has an SSE $= 19.6575$. This difference is probably insignificant and worth the sacrifice for a model that is more consistent and interpretable.

The treed regression models can be improved by applying the base algorithm recursively to create a model with four terminal nodes. The monotone treed regression model given in Figure 9 further splits the model in Figure 8 using ptratio and crim. In addition to SSE, consider the model complexity in judging a model's performance. The

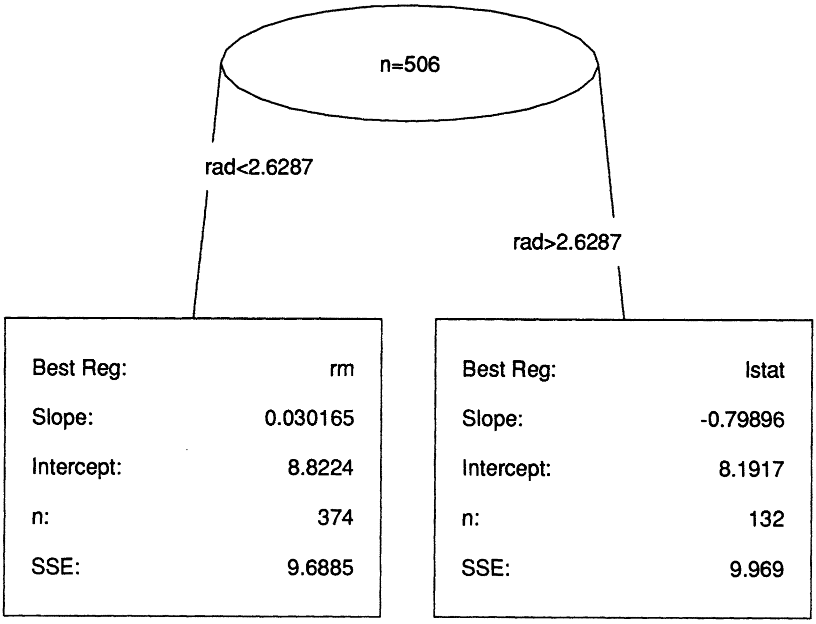


Figure 8. Monotone treed regression model of the log of the median value of owner-occupied homes using 13 explanatory variables measured on Boston area census tracts in 1970 from an example in Belsley, Kuh, and Welsch (1980).

monotone treed regression model in Figure 9 yields an SSE = 14.3215 while estimating eight regression parameters (two at each of the four terminal nodes) and three split parameters in the tree structure.

For comparison purposes, the linear regression model constructed from all 13 available explanatory variables has an SSE = 16.37823 and estimates 14 regression parameters. Belsley, Kuh, and Welsch (1980) commented that the condition number is of moderate size, but they did not pursue the collinearity in this data set. However, the regression coefficient associated with `indus` is opposite in sign from the expected relationship with Y, which could be due to collinearity.

The CART model in Figure 4 has an SSE = 13.17522 and estimates 14 regression parameters, a mean at each of the 14 terminal nodes, and 13 split parameters in the tree structure. Monotonicity is not imposed and is most likely violated throughout.

The MARS model has an extremely small SSE = 7.381841. However, this is a significantly more complex model than was fitted by treed regression, linear regression, or CART. Sixteen knot locations are estimated (which are comparable to the number of split parameters) along with 25 regression parameters (one for each of the 24 basis functions and β_0). Also, it is difficult to discern any neighborhood effects from this complicated functional structure. Furthermore, although the MARS model is continuous, it is not clear if the desired monotonicity is maintained.

Treed regression provides a model that compares favorably to other regression methodologies. By incorporating a small tree structure it is possible to model the neighborhood effects on housing prices that are not present in a linear regression model. By imposing monotonicity in some of the explanatory variables, the treed regression model

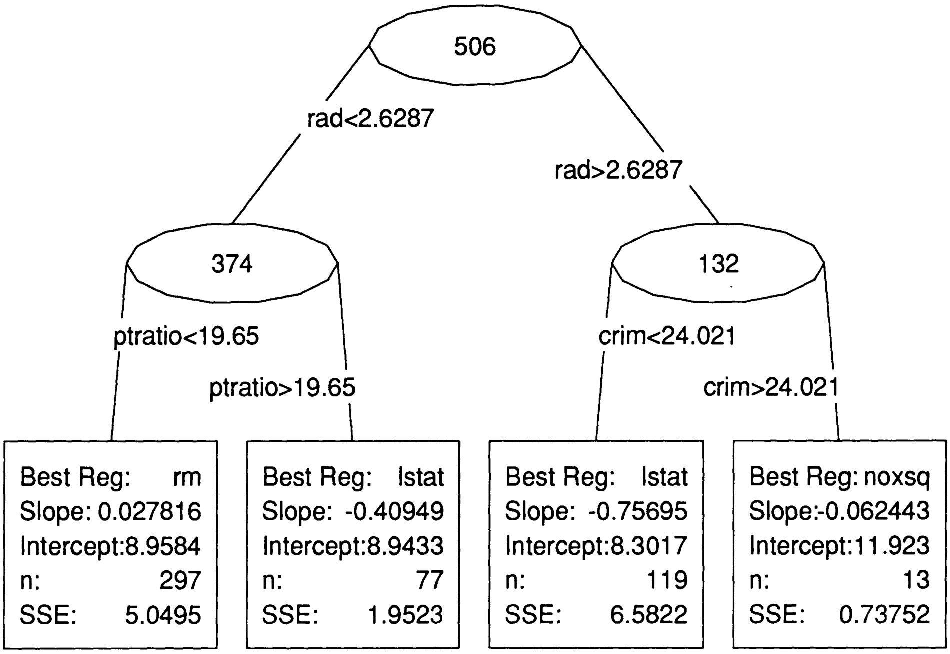


Figure 9. Monotone treed regression model of level two (four terminal nodes) of the log of the median value of owner-occupied homes using 13 explanatory variables measured on Boston area census tracts in 1970 from an example in Belsley, Kuh, and Welsch (1980).

maintains expected relationships which, if ignored, decrease the practical application and interpretation of the model.

6. SUMMARY

The treed regression methodology offers a powerful modeling technique to the researcher. Treed regression is seen to possess many of the desirable qualities of both CART and linear regression models. The tree structure, as in CART, generates a segmentation of the data set. Such segmentations, insofar as they are interpretable, are of interest in their own right. The more sophisticated leaf models of treed regression generally permit less complex, more interpretable segmentations.

The base algorithm for treed regression identifies a split using an independent variable and the best simple linear regression models for each subset created by the split. This algorithm is computationally intense because all possible splits of the observations and all possible regressions must be computed. Trees of any desired depth can be created by applying the base algorithm recursively.

Monotonicity in some or all of the independent variables is difficult to verify in even small tree structures. Therefore, if expected relationships between the dependent and independent variables are to be maintained, the base algorithm must be changed. The necessary modifications are described in two steps. First, the algorithm is modified to maintain within node monotonicity by considering only those regression models whose slope agrees with the monotonicity constraint. The algorithm is then further modified to impose between-node monotonicity by considering the leaves that would be used by making the observation "worse" in terms of the expected relationships with the independent variables.

[Received March 1995. Revised October 1995.]

REFERENCES

Alexander, W. P., and Grimshaw, S. D. (1994), "Computational Efficiency in Treed Regression Using the QR Decomposition," *BYU Statistics Department Report Series*, SD-053-R, Brigham Young University.

Belsley, D. A., Kuh, E., and Welsch, R. E. (1980), *Regression Diagnostics*, New York: John Wiley.

Breiman, L., and Meisel, W. S. (1976), "General Estimates of the Intrinsic Variability of Data in Nonlinear Regression Models," *Journal of the American Statistical Association*, 71, 301–307.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984), *Classification and Regression Trees*, Belmaont, CA: Wadsworth.

Chaudhuri, P., Huang, M. C., Loh, W. Y., and Yao, R. (1994), "Piecewise-Polynomial Regression Trees," *Statistica Sinica*, 4, 143–167.

Duarte, C. M., and Kalff, J. (1990), "Patterns in the Submerged Macrophyte Biomass of Lakes and the Importance of the Scale of Analysis in the Interpretation," *Canadian Journal of Fisheries and Aquatic Sciences*, 47, 357–363.

Friedman, J. M. (1991), "Multivariate Adaptive Regression Splines," *The Annals of Statistics*, 19, 1–141.

Karalič, A. (1992), "Employing Linear Regression in Regression Tree Leaves," in *Proceedings of the Tenth European Conference on Artificial Intelligence*, ed. B. Neumann, New York: John Wiley, pp. 440–441.

Seber, G. A. F. (1977), *Linear Regression Analysis*, New York: John Wiley.

Zacks, S. (1991), "Detection and Change-Point Problems," in *Handbook of Sequential Analysis*, eds. B. K. Ghosh and P. K. Sen, New York: Marcel Dekker, pp. 531–562.